

Министерство образования и науки Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»

Кафедра информатики и программирования

**«Проблема выбора языка при изучении объектно-
ориентированного программирования»**
АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

Студента 4 курса 461 группы

направления 44.03.01 Педагогическое образование (профиль Информатика)

факультета компьютерных наук и информационных технологий

Епрынцева Дмитрия Викторовича

Научный руководитель

старший преподаватель

Е.М. Черноусова

Зав. кафедрой

доцент кафедры, к. ф.м.н

М.В. Огнева

Саратов 2019

ВВЕДЕНИЕ

Мнения о подходе к изучению программирования в школьном курсе информатики очень сильно разнятся: некоторые утверждают, что стоит оставить программирование и начало алгоритмизации для профильных классов, а на базовом уровне изучать только пакет офисных программ; другие же утверждают, что необходимо углубляться в операционные системы и несколько языков программирования с их разного рода парадигмами. Поиск золотой середины не такая простая задача, как может показаться на первый взгляд.

Для начала нужно определить цель, которую учитель хочет достичь. Научить своих детей мыслить логически и алгоритмически? Познакомить с персональными компьютерами на базовом уровне, научить пользоваться интернетом, текстовыми и графическими редакторами? Или заложить базовые знания, которые пригодятся будущему IT-специалисту?

Некоторые преподаватели считают, что не столь важно, какой язык программирования будет использоваться на уроках информатики; важно научить не языку программирования, а методам программирования и системному подходу решения задач. Необходимо развивать алгоритмическое мышление, которое в будущем поможет при углубленном изучении программирования.

Действительно ли не столь важно, какой язык программирования будет использоваться на уроках? У каждого учителя информатики есть персональный список требований к языку, который будет использоваться на практических занятиях. В эти требования может входить, например, это: простота, интуитивный синтаксис, наличие так называемой справки, наличие версий под различные платформы и операционные системы.

Казалось бы, школьная программа не предполагает подготовку высококлассных специалистов и все базовые понятия можно давать на Паскале, который завоевал популярность как язык, с которого стоит начать знакомство с алгоритмизацией. Ученику достаточно знать основы бинарной логики,

основные конструкции: условия, циклы, подпрограммы, а все эти понятия присутствуют в большинстве языков программирования.

С иной стороны IT-технологии развиваются очень быстро, появляются новые учебные платформы, появляются новые языки программирования, появляются новые технологии программирования. Современному программисту приходится постоянно развиваться, изучать нечто новое для того, чтобы поспевать за всеми трендами, которые постоянно меняются в условиях непрерывного развития. Именно поэтому стоит посмотреть вперед, в перспективу и, исходя из этой перспективы подходить к обучению программированию.

Да, развитие не стоит на месте, но есть вещи, которые никогда не меняются, они остаются неизменными до сих пор. Именно этот базис необходимо изначально передавать учащимся.

В программировании существует несколько парадигм: модульная, процедурная и объектно-ориентированная. В школах, как правило, дается процедурный подход к программированию, когда объектно-ориентированный — почти никогда.

Выбор языка программирования играет очень важную роль, ведь от данного выбора зависит все остальное. Такие вещи, например, как методика изучения курса, его содержание, последовательность.

Целью данной работы является помочь учителю выбрать язык программирования, с которого можно начать знакомство с объектно-ориентированным программированием (ООП). В настоящее время, как уже и говорилось, объектно-ориентированное программирование затрагивается в школах крайне редко и, собственно, поэтому встает вопрос о том, на каком языке лучше подать его ученикам.

Для процедурного программирования можно использовать Паскаль, который имеет строгую и статическую типизацию, семантику. Это отличное начало для учеников, но что дальше?

Объектно-ориентированное программирование — основа основ современных языков программирования. И с этой основой необходимо знакомиться еще в школе, чтобы ученики, которые приходят в высшие учебные заведения, уже имели представление об объектно-ориентированном программировании. Необходимо заметить, что данное утверждение применимо только к ученикам, которые хотят развиваться в данном направлении.

В нашем случае, необходимо выбрать язык, который подойдет для того, чтобы начать знакомство с объектно-ориентированным программированием. Как это сделать, если не путем сравнения, выявления плюсов и минусов каждого из языков.

Для достижения поставленной цели необходимо решить следующие задачи:

- рассмотреть особенности объектно-ориентированного программирования в различных языках (PascalABC, Python, C++);
- проанализировать возможные сложности для учащихся при переходе на новый язык при изучении данной парадигмы программирования;
- провести анализ изложения данной темы в учебно-методической литературе;
- провести опрос студентов факультета компьютерных наук и информационных технологий.

Основное содержание работы

Выпускная квалификационная работа содержит в себе пять глав, в одной из которых проводится сравнительный анализ трех языков программирования: Паскаль, С++ и Питон. Последняя глава является анализом опроса, проведенного среди студентов первого курса факультета компьютерных наук и информационных технологий Саратовского Национального Исследовательского Университета имени Н. Г. Чернышевского.

Такой язык программирования как Паскаль, занимает особое место в обучении детей началам алгоритмизации и программированию. Программы, которые пишутся на этом языке выглядят очень структурированно. Переменные не берутся из ниоткуда. В начале программы существует раздел, который отвечает за объявление переменных и их типов. Четко видно «тело» программы, заключенное в `begin` и `end`.

Строгая типизация и семантика языка заставляют раскладывать программный код по блокам, которые в совокупности создают структурированный код. Паскаль заставляет понять основу алгоритмизации, ее принцип. Паскаль не имеет возможности сильно запутать ребенка, ведь он строго типизированный. Он не дает смешивать целые числа с вещественными, он запрещает смешивать числовые переменные с текстовыми. Паскаль исключает возможность зарождения такой мысли, которая приведет к неправильному представлению о методах хранения данных в памяти компьютера.

Уже говорилось о том, что Паскаль был создан как раз для обучения программированию. Данные слова подтверждаются действиями компилятора, который старается оповестить о как можно большем количестве ошибок. Ведь чем больше своих ошибок дети увидят, и чем больше поймут сообщений, которые повествуют об их ошибках, тем больше знаний они получат.

Казалось бы, проблем нет, но изучая программирование в школе, дети привыкают к процедурному программированию. Отчего первая встреча с объектно-ориентированным программированием становится чем-то страшным

и сложным. Большая часть студентов сталкивается с трудностями в понимании данной темы. Следует предположить: чем раньше дети столкнутся с ООП, тем проще им будет в высшем учебном заведении освоить данный материал, не испытывая особых трудностей в понимании концепции объектно-ориентированного программирования.

Следующим языком для рассмотрения возьмем довольно известный язык программирования — C++. Данный язык предлагается в качестве одного из пяти языков на выбор для решения задач второй части, поэтому его также необходимо рассмотреть.

Как упоминалось ранее, одним из минусов изучения объектно-ориентированного программирования на языке PascalABC.Net было отсутствие энтузиазма у учеников продолжать обучение на том же языке, когда есть возможность разобрать новую тему и положить начало изучению нового языка программирования, который в дальнейшем может пригодиться куда больше, чем PascalABC.Net. Бесспорно, начало алгоритмизации стоит изучать с помощью того же Паскаля, так как программный код четко структурирован, обладает наиболее понятными ключевыми словами, которые способствуют лучшему запоминанию материала.

Конструктор в C++ выполняет ту же функцию, что и в других языках программирования. Функция-член, единственная работа которой заключается в инициализации объекта класса, называется конструктором.[4] Заметим, что имя конструктора обязательно должно совпадать с именем класса, в котором он объявляется.

Выделить стоит такую вещь, как указатель `this`. В данном языке, в отличие от Питона, его явное использование необязательно. В C++ указатель `this` используется неявно автоматически, когда дело касается полей класса.

Указатель `this` — это указатель на адрес объекта класса, при этом он является скрытым первым параметром любого метода класса (кроме статических методов), а типом указателя выступает имя класса. И еще кое-что из теории:

- `this` является зарезервированным словом C++;

- мы можем использовать `this` явно в методах-элементах класса;
- а вот явно объявить, инициализировать либо изменить указатель `this`, возможности нет;

На сегодняшний день особой популярностью пользуется такой язык программирования как Python. Данный язык был создан более 20 лет назад, но и по сей день пользуется популярностью в качестве языка, на котором можно преподавать объектно-ориентированное программирование. На данном языке можно разрабатывать больше количество программ: серверные, клиентские, различные сайты, мобильные приложения и многое другое. Это сразу же подводит к одному из первых плюсов, которые содержит в себе данный язык программирования. Этим плюсом является то, что на рынке труда Python-программисты востребованы, поэтому изучение данного языка не пройдет в пустоту.

Python — это полноценный язык программирования. В его распоряжении имеется целый ряд парадигм: функциональная парадигма, модульная, процедурная и, что самое главное для данной работы, объектно-ориентированная. Важно заметить, что Python распространяется по бесплатной модели. Это означает, что его можно спокойно установить на персональные компьютеры в школе, оставляя проблему оплаты в стороне.

Следующим очевидным плюсом является то, что Python является интерпретируемым языком. Данный факт говорит о том, что можно избежать ненужный код при генерации новых идей. К тому же, интерпретируемость повышает обнаружение ошибок, так как сразу станет понятно в каком месте находится ошибка. Следующим очевидным плюсом является то, что данный язык обладает достаточно ясным синтаксисом, отчего читаемость кода повышается, так как в нем нет огромной нагрузки вспомогательными элементами. Вместо этого Python предлагает программисту делать отступы, тем самым повышая читаемость кода.

У данного языка есть как плюсы, так и минусы. Динамическая типизация Python'a может привести к путанице. Для опытного программиста

это будет скорее плюсом, чем минусом, но с обучающим данный факт может сыграть злую шутку. Ведь, если Паскаль и С++ предлагают строгую типизацию, которая может помочь избежать путаницы с хранением данных; и заставляет следить за типами переменных, тем самым способствуя лучшему пониманию того, как происходит хранение данных в памяти компьютера.

Резюмируя, можно сказать, что Питон имеет ряд преимуществ и лазеек, которые могут облегчить изучение объектно-ориентированного программирования, но, если этот язык первый у ребенка, то может быть достаточно проблематичным переход на другие языки программирования. В большинстве своем, это упирается в динамическую типизацию, которая является как плюсом, так и минусом, с которым ничего не поделаешь.

Из очевидных плюсов можно отметить легкий синтаксис, который свободно читается и изучается. Большая часть слов интуитивна, блоки кода регулируются табуляцией, заменяя собой фигурные скобки или ключевые слова вроде `begin` и `end`. Все базовые конструкции мало чем отличаются от того же Паскаля или С++, в некоторых школах и вовсе программирование преподается на Питоне, отчего возможностей познакомить детей с объектно-ориентированным программированием становится все больше, ведь в отличие от Паскаля, Питону не нужно давать такую тему, как записи. Все, что нужно для изучения объектно-ориентированного программирования на Питоне это основные базовые конструкции и четкое понимание того, что они делают.

Определенной проблемой в выборе языка программирования является желание учеников и возможности учителя. Если учитель не обладает знаниями, которые необходимы для того, чтобы обучать программированию на том или ином языке, то хорошего из этого ничего не выйдет. Но, как правило, базовые конструкции вроде циклов, условных операторов и т.д. больше строятся на понимании самого алгоритма, нежели на синтаксисе языка. Короче говоря, если учитель понимает принцип работы данных конструкций, то разучить синтаксис не составит никакого труда.

Был проведен опрос среди студентов факультета компьютерных наук и информационных технологий Саратовского Национального Исследовательского Государственного Университета имени Н.Г. Чернышевского, результаты которого очень хорошо отражают желания учеников и действительность, в которой они проходили школьный курс информатики.

ЗАКЛЮЧЕНИЕ

Объектно-ориентированное программирование является одним из важнейших подходов к программированию, которые существуют на сегодняшний день. Обучать ему в школьном курсе информатики или нет, решать только учителям. В профильных классах необходимо углубленно изучать предмет и уделять большое внимание программированию, которое в дальнейшем будет занимать большую часть жизни учеников.

Проблема выбора языка остается открытой по сей день. Ведь этот выбор зависит не только от желания учеников, но и от уровня подготовки преподавателя, который и будет доносить до детей знания, что помогут им в высших учебных заведениях, а потом и на выбранной работе. Уже сейчас в Саратовском национальном исследовательском государственном университете имени Н.Г. Чернышевского на направлении, отвечающем за подготовку учителей информатики, изучают объектно-ориентированное программирование.

Если касаться каждого из рассмотренных языков, то каждый из них обладает рядом плюсов и минусов, большая часть которых, опять же, скорее субъективна, нежели объективна. Все три, например, объединяет то, что каждый из рассмотренных языков имеет бесплатную среду разработки. Выделить, разве что, можно строгую или динамическую типизацию, которая отличает Паскаль и С++ от Питона. В остальном, выбор между скобками, ключевыми словами или табуляцией целиком и полностью зависит от учителя. Дело не только в наличии семантики или типизации. Каждый из языков обладает рядом подводных камней, с которыми так или иначе придется столкнуться.